SPYWOLF

Security Audit Report

Audit prepared for

PANIC APE

Completed on

March 30, 2024

KEY RESULTS

Cannot mint new tokens	*
Cannot pause trading (honeypot)	Passed
Cannot blacklist an address	Passed
Cannot raise taxes over 25%?	Passed
No proxy contract detected	Passed
Not required to enable trading	Passed
No hidden ownership	Passed
Cannot change the router	Passed
No cooldown feature found	Passed
Bot protection delay is lower than 5 blocks	Passed
Cannot set max tx amount below 0.05% of total supply	Passed
The contract cannot be self-destructed by owner	Passed

For a more detailed and thorough examination of the heightened risks, refer to the subsequent parts of the report.

N/A = Not applicable for this type of contract

*New tokens can be minted via the factory contract

OVERVIEW

This goal of this report is to review the main aspects of the project to help investors make an informative decision during their research process.

You will find a a summarized review of the following key points:

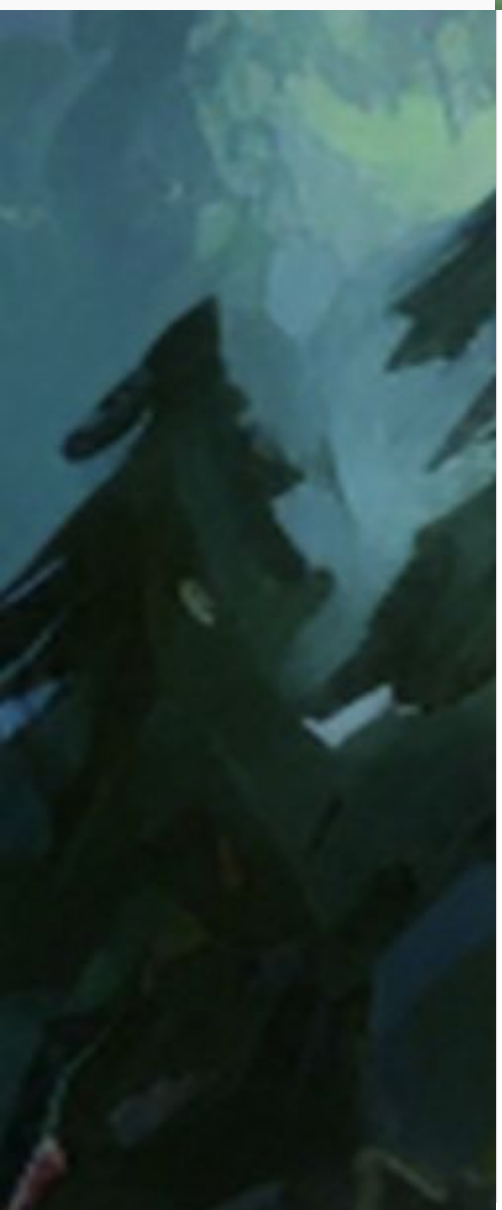
- ✔ Contract's source code
- ✔ Owners' wallets
- ✔ Tokenomics
- ✔ Team transparency and goals
- ✔ Website's age, code, security and UX
- ✔ Whitepaper and roadmap
- ✔ Social media & online presence

The results of this audit are purely based on the team’s evaluation and does not guarantee nor reflect the projects outcome and goal

- SPYWOLF Team -

TABLE OF CONTENTS

PANIC APE



PROJECT DESCRIPTION

According to their whitepaper:

The mission of the PAPE Token protocol is to create a decentralized, open, and fair financial system that distributes the participation and influence of all participants.

The protocol aims to provide a transparent and accessible mechanism for liquidity provision, allowing users to participate in sustainable and reliable economic growth.

Release Date: Launching March, 2024

Category: DeFi

PANIC APE CONTRACT INFO

Token Name
PanicApeToken

Symbol
PAPE

Contract Address
0x57B478bbd84eCD5D2F1a98f3D556e1fB2d51dcE1

Network
Binance Smart Chain

Language
Solidity

Deployment Date
March 28, 2024

Contract Type
Token without taxes

Total Supply
420,000,000

Status
Not launched

TAXES

Buy Tax
none

Sell Tax
none

*Taxes cannot be changed

Our Contract Review Process

The contract review process pays special attention to the following:

- Testing the smart contracts against both common and uncommon vulnerabilities
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Blockchain security tools used:

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

TOKEN TRANSFERS STATS

Transfer Count	2
Uniq Senders	2
Uniq Receivers	2
Total Amount	420100000.00000006 PAPE
Median Transfer Amount	420000000 PAPE
Average Transfer Amount	210050000.00000003 PAPE
First transfer date	2024-03-28
Last transfer date	2024-03-28
Days token transferred	1

SMART CONTRACT STATS

Calls Count	7
External calls	3
Internal calls	4
Transactions count	5
Uniq Callers	1
Days contract called	1
Last transaction time	2024-03-28 10:15:24 UTC
Created	2024-03-28 04:41:14 UTC
Create TX	0x37a5fbf0eda5269a4163f118d08b2b716187a19e248cad76f463742a0dd5c53e
Creator	0xACDd59B1F5DD345E15A5B8515F977531949272A8

FEATURED WALLETS

Deployer address	0xACDd59B1F5DD345E15A5B8515F977531949272A8
Marketing fee receiver	N/A
LP address	**Pancakeswap:** 0xA5ec0a510Db42C7B3E1a5103a46B5B6c238744d0

TOP 3 UNLOCKED WALLETS

99.97%	0xACDd59B1F5DD345E15A5B8515F977531949272A8 Same as deployer
N/A	
N/A	

VULNERABILITY ANALYSIS

ID	Title	
SWC-100	Function Default Visibility	**Passed**
SWC-101	Integer Overflow and Underflow	**Passed**
SWC-102	Outdated Compiler Version	**Passed**
SWC-103	Floating Pragma	**Passed**
SWC-104	Unchecked Call Return Value	**Passed**
SWC-105	Unprotected Ether Withdrawal	**Passed**
SWC-106	Unprotected SELFDESTRUCT Instruction	**Passed**
SWC-107	Reentrancy	**Passed**
SWC-108	State Variable Default Visibility	**Passed**
SWC-109	Uninitialized Storage Pointer	**Passed**
SWC-110	Assert Violation	**Passed**
SWC-111	Use of Deprecated Solidity Functions	**Passed**
SWC-112	Delegatecall to Untrusted Callee	**Passed**
SWC-113	DoS with Failed Call	**Passed**
SWC-114	Transaction Order Dependence	**Passed**
SWC-115	Authorization through tx.origin	**Passed**
SWC-116	Block values as a proxy for time	**Passed**
SWC-117	Signature Malleability	**Passed**
SWC-118	Incorrect Constructor Name	**Passed**

VULNERABILITY ANALYSIS

ID	Title	
SWC-119	Shadowing State Variables	Passed
SWC-120	Weak Sources of Randomness from Chain Attributes	Passed
SWC-121	Missing Protection against Signature Replay Attacks	Passed
SWC-122	Lack of Proper Signature Verification	Passed
SWC-123	Requirement Violation	Passed
SWC-124	Write to Arbitrary Storage Location	Passed
SWC-125	Incorrect Inheritance Order	Passed
SWC-126	Insufficient Gas Griefing	Passed
SWC-127	Arbitrary Jump with Function Type Variable	Passed
SWC-128	DoS With Block Gas Limit	Passed
SWC-129	Typographical Error	Passed
SWC-130	Right-To-Left-Override control character (U+202E)	Passed
SWC-131	Presence of unused variables	Passed
SWC-132	Unexpected Ether balance	Passed
SWC-133	Hash Collisions With Multiple Variable Length Arguments	Passed
SWC-134	Message call with hardcoded gas amount	Passed
SWC-135	Code With No Effects	Passed
SWC-136	Unencrypted Private Data On-Chain	Passed

VULNERABILITY ANALYSIS

NO ERRORS FOUND

MANUAL
CODE REVIEW

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time.

We categorize these vulnerabilities by 4 different threat levels.

THREAT LEVELS

High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium Risk

Issues on this level are critical to the smart contract's performance, functionality and should be fixed before moving to a live environment.

Low Risk

Issues on this level are minor details and warning that can remain unfixed.

Informational

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

FOUND THREATS

⚠ High Risk

No high risk-level threats found in this contract.

⚠ Medium Risk

No medium risk-level threats found in this contract.

⚠ Low Risk

No low risk-level threats found in this contract.

FOUND THREATS

Informational

New tokens can be minted via the factory contract.
Newly minted tokens can be up to total supply of 2,100,000,000.
Current total token supply is 420,000,000.
Factory contract cannot be changed.
Current factory contract is set at address:
0x6488A6598c11d8F4320a985B2749Ab7410a402A5

Note: Factory contract is not in the scope of the current audit.

```
address public factory;
uint256 public maxSupply = 2100000000 * 10 ** 18;
constructor (address _factory ) ERC20("PanicApeToken", "PAPE") {
    factory = _factory;
    ............
 }

function mint(address to, uint256 amount) external {
    require(factory == msg.sender, "PanicApeToken: caller is not factory!");

    require((totalSupply() + amount) <= maxSupply, "PanicApeToken: total supply exceeds limit!");
    _mint(to, amount);
}

function _mint(address account, uint256 amount) internal virtual {
    require(account != address(0), "ERC20: mint to the zero address");

    _beforeTokenTransfer(address(0), account, amount);

    _totalSupply += amount;
    _balances[account] += amount;
    emit Transfer(address(0), account, amount);

    _afterTokenTransfer(address(0), account, amount);
}
```

PANIC APE FACTORY

Token Name
N/A

Symbol
N/A

Contract Address
0x6488A6598c11d8F4320a985B2749Ab7410a402A5

Network
Binance Smart Chain

Language
Solidity

Deployment Date
March 28, 2024

Contract Type
Token without taxes

Total Supply
N/A

Status
Launched

TAXES

Buy Tax
UP TO 10%

Sell Tax
none

*Taxes cannot be changed

Our Contract Review Process

The contract review process pays special attention to the following:

- Testing the smart contracts against both common and uncommon vulnerabilities
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Blockchain security tools used:

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

FOUND THREATS

Informational

Owner can initiate contract and set the token once.
Contract is currently initiated.
Token set is:
0x57B478bbd84eCD5D2F1a98f3D556e1fB2d51dcE1

```
function init(address _token) external onlyOwner {
    require(!hasInit, "PanicApeFactory: has been init!");
    token = _token;
    hasInit = true;

    IUniswapV2Router01 router = IToken(token).router();
    IERC20(token).approve(address(router), MAX_VALUE);
}
```

Multi sig wallets cannot participate as referrers.

```
function _referrerReward(address miner, uint256 value) private {
    address referrer = miners[miner].referrer;

    for(uint i = 0; i < REF_FEE.length; i++){
        uint256 refRewards = value.mul(REF_FEE[i]).div(PERCENT_DIVIDER);
        _ethTransfer(referrer, refRewards);
        miners[referrer].referralRewards = miners[referrer].referralRewards + refRewards;
        referrer = miners[referrer].referrer;
        if(address(0) == referrer){
            break;
        }
    }
}

function _ethTransfer(address to, uint256 value) private {
    payable(to).transfer(value);
}
```

FOUND THREATS

Informational

Owner can set swap slip up to 10.
Swap slip is used as multiplier for tokens mint amount when users uses the withdraw() function.

```
function setSwapslip(uint256 _swapslip) external onlyOwner{
    require(_swapslip > 2 && _swapslip < 10, "PanicApeFactory: swapslip too low!");
    swapslip = _swapslip;
}

 function _swapToExactETH(uint256 value) private {
    IUniswapV2Router01 router = IToken(token).router();
    address[] memory path = new address[](2);
    path[0] = token;
    path[1] = weth;
    uint256 amount = getTokenAmount(value) * swapslip;
    _mintToken(address(this), amount);

    router.swapTokensForExactETH(
            value,
            amount,
            path,
            msg.sender,
            block.timestamp + 1 minutes
    );
}

function withdraw(uint256 tokenId, bool onlyToken) external nonReentrant {
    require(tokenIds[tokenId], "PanicApeFactory: invalid tokenId!");
    require(msg.sender == panicApe.ownerOf(tokenId), "PanicApeFactory:you are not the owner!");
    SharedStructs.Ape memory ape = panicApe.apes(tokenId);

    require(ape.refundTime < block.timestamp, "PanicApeFactory:It is not time to withdraw yet!");

    _claim(msg.sender);

    _subInvest(msg.sender, ape.value, ape.daily);
    if(onlyToken){
        uint256 amount = getTokenAmount(ape.value);
        _mintToken(msg.sender, amount);
    }else{
        _swapToExactETH(ape.value);
    }

    panicApe.burnApe(tokenId);
    tokenIds[tokenId] = false;
}
```

FOUND THREATS

Informational

Owner can close items sells.
Once closed it cannot be turned on again.

```
function closeSell() external onlyOwner {
    closeItemSell = true;
}

function sellItems(ItemStruct calldata itemStruct, uint8 _v, bytes32 _r, bytes32 _s) external nonReentrant{
    require(!ids[itemStruct.id], "PanicApeFactory: id already used!");
    require(itemStruct.amount <= 5, "PanicApeFactory: amount must be less than 5!");
    require(!closeItemSell, "PanicApeFactory: Item cannot be sold!");
    verifyItem(itemStruct, _v, _r, _s);

    uint256 value = getItemValue(itemStruct.itemId).mul(itemStruct.amount);
    uint256 tokenAmount = getTokenAmount(value);
    _mintToken(itemStruct.owner, tokenAmount);

    ids[itemStruct.id] = true;
    emit SellItems(itemStruct.owner, itemStruct.id, itemStruct.itemId, itemStruct.amount);
}
```

FOUND THREATS

Informational

Owner can change the signer address which is used for verification in updateItem() and sellItems functions().

```
function setSigner(address _signer) external onlyOwner{
    signer = _signer;
}

function updateItem(SharedStructs.UpdateStruct calldata updateStruct, uint8 _v, bytes32 _r, bytes32 _s) external nonReentrant{
    require(!ids[updateStruct.id], "PanicApeFactory: id already used!");
    require(tokenIds[updateStruct.tokenId], "PanicApeFactory: invalid tokenId!");
    require(msg.sender == panicApe.ownerOf(updateStruct.tokenId), "PanicApeFactory: invalid owner!");
    verifyUpdate(updateStruct, _v, _r, _s);
.....................
}

function verifyUpdate(SharedStructs.UpdateStruct calldata updateStruct, uint8 _v, bytes32 _r, bytes32 _s) public view {
    require(msg.sender == updateStruct.owner && updateStruct.expired > block.timestamp, "PanicApeFactory: invalid data!");
    address _signer = verifyMessage(keccak256(abi.encode(updateStruct)), _v, _r, _s);
    require(signer == _signer, "PanicApeFactory: wrong signature!");
.....................
}

function sellItems(ItemStruct calldata itemStruct, uint8 _v, bytes32 _r, bytes32 _s) external nonReentrant{
    require(!ids[itemStruct.id], "PanicApeFactory: id already used!");
    require(itemStruct.amount <= 5, "PanicApeFactory: amount must be less than 5!");
    require(!closeItemSell, "PanicApeFactory: Item cannot be sold!");
    verifyItem(itemStruct, _v, _r, _s);
    .....................
}

function verifyItem(ItemStruct calldata itemStruct, uint8 _v, bytes32 _r, bytes32 _s) public view{
    require(msg.sender == itemStruct.owner && itemStruct.expired > block.timestamp, "PanicApeFactory: invalid data!");
    address _signer = verifyMessage(keccak256(abi.encode(itemStruct)), _v, _r, _s);
    require(signer == _signer, "PanicApeFactory: wrong signature!");
}
```

PANIC APE ITEMS

Token Name	Symbol
N/A	N/A

Contract Address
0xfF3e4709169C76f43d9d3137a6c39dFC0bc37baC

Network	Language
Binance Smart Chain	Solidity

Deployment Date	Contract Type
March 28, 2024	NFT

Total Supply	Status
N/A	Launched

TAXES

Buy Tax
none

Sell Tax
none

*Taxes cannot be changed

Our Contract Review Process

The contract review process pays special attention to the following:

- Testing the smart contracts against both common and uncommon vulnerabilities
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Blockchain security tools used:

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

FOUND THREATS

⚠ Medium Risk

Owner can change factory address at any time.
Via the factory address, new NFT items can be minted via burnApe function.
Via the factory address, already existing NFT items can be burned.
Via the factory address, already existingNFT items stats can be updated.

```
function setFactory(address _factory) external onlyOwner{
    factory = _factory;
}

function mintApe(
        address owner, uint256 tokenId, uint256 index,
        uint256 value, uint256 daily, uint256 refundTime) external {
    require(factory == _msgSender(), "PanicApe: caller is not the factory!");
    _mint(owner, tokenId);

    apes[tokenId].creator = owner;
    apes[tokenId].value = value;
    apes[tokenId].index = index;
    apes[tokenId].daily = daily;
    apes[tokenId].refundTime = refundTime;
    emit MintApe(owner, tokenId, index, value, daily, refundTime);
}

function burnApe(uint256 tokenId) external{
    require(factory == _msgSender(), "PanicApe: caller is not the factory!");
    _burn(tokenId);
}

function updateItem(SharedStructs.UpdateStruct calldata updateStruct) external {
    require(factory == _msgSender(), "PanicApe: caller is not the factory!");
    SharedStructs.Ape memory ape = apes[updateStruct.tokenId];

    uint8 oldPartRate = getPartRate(ape.part1, ape.part2, ape.part3);
    uint8 partRate = getPartRate(updateStruct.part1, updateStruct.part2, updateStruct.part3);

    uint256 addDaily = ape.value.mul(partRate - oldPartRate).div(PERCENT_DIVIDER);
    apes[updateStruct.tokenId].daily = ape.daily + addDaily;

    apes[updateStruct.tokenId].part1 = updateStruct.part1;
    apes[updateStruct.tokenId].part2 = updateStruct.part2;
    apes[updateStruct.tokenId].part3 = updateStruct.part3;

    emit UpdateItem(
        updateStruct.owner, updateStruct.tokenId, updateStruct.id,
        updateStruct.part1, updateStruct.part2 updateStruct.part3, addDaily
    );
}
```

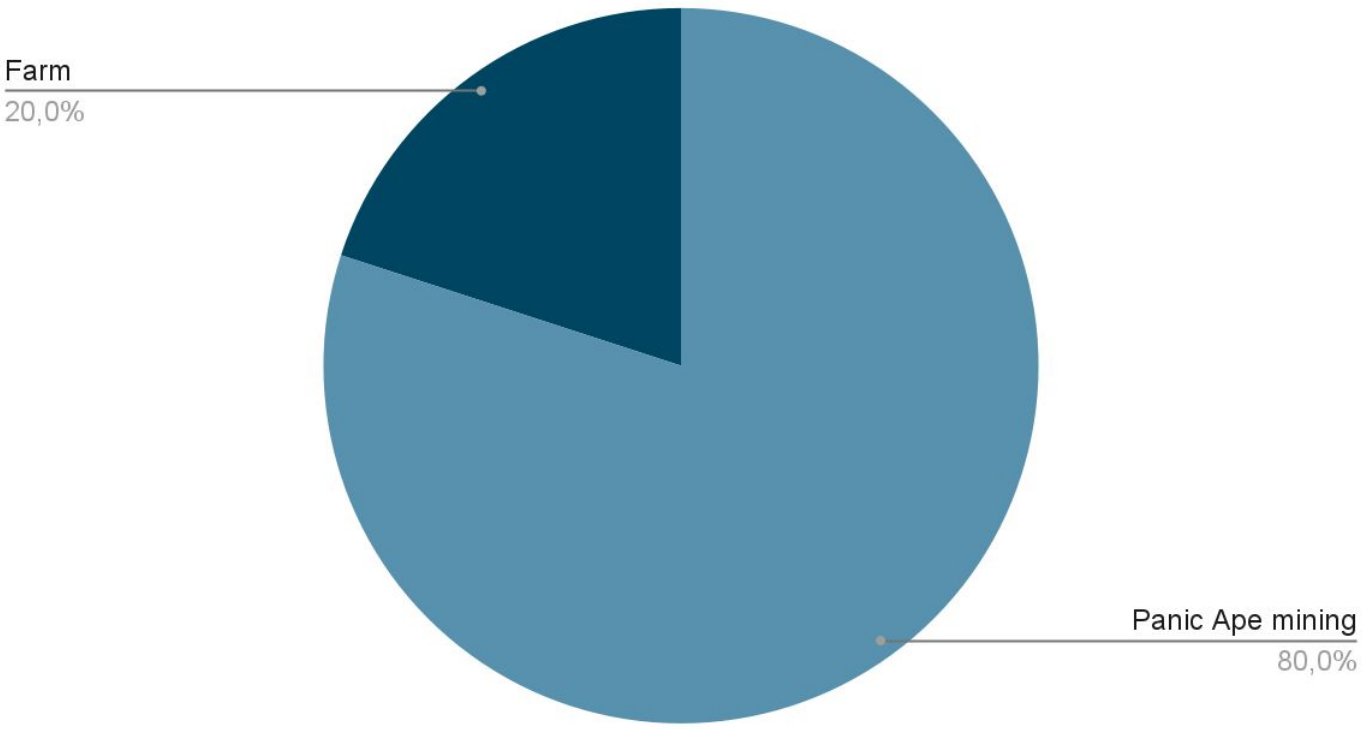
NFT’s and their stats are directly linked with the amounts of PAPE token that can be minted by user.

TOKENOMICS

The following tokenomics are based on the project's whitepaper and/or website:

- 80% - Panic Ape mining
- 20% - Farm

Tokens distribution



WEBSITE

Website URL
https://www.panicape.com/

Domain Registry
https://www.hostinger.com

Domain Expiration
2025-03-16

Technical SEO Test
Passed

Security Test
Passed. SSL certificate present

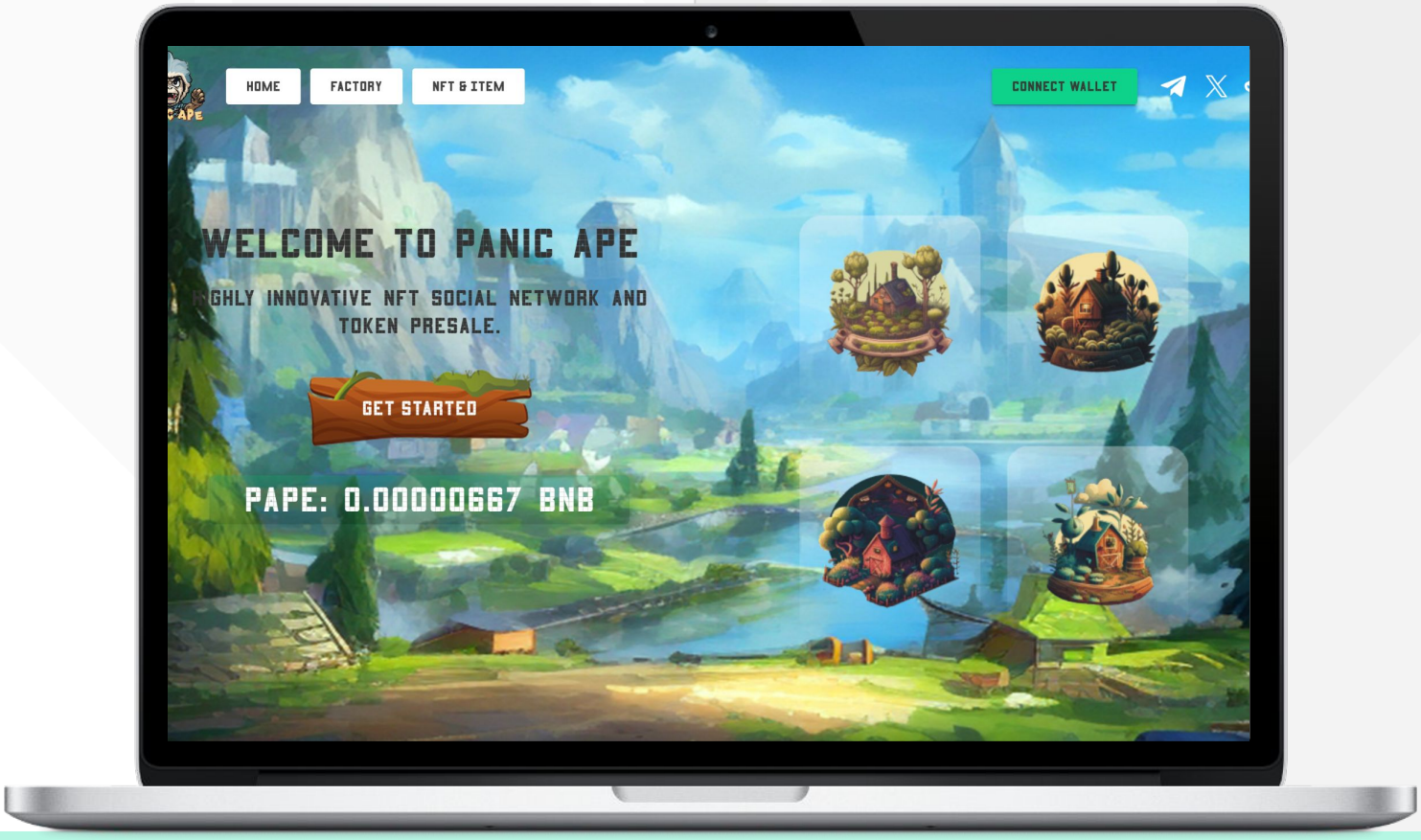
Design
Very nice overall design with appropriate color scheme and graphics.

Content
The information helps new investors understand what the product does right away. No grammar mistakes found.

Whitepaper
Well written, explanatory.

Roadmap
Yes, goals set with time frames.

Mobile-friendly?
Yes



panicape.com

SOCIAL MEDIA

& ONLINE PRESENCE

ANALYSIS

Project's social media pages are active

Twitter's X

@panicapegroup

- 617 followers
- Posts frequently
- Active

Discord

- Not available

Telegram

@panicapeofficial

- 294 members
- Few active members
- Active mods

Medium

- Not available

ABOUT US

We are a growing crypto security agency offering audits, KYCs and consulting services for some of the top names in the crypto industry.

- **OVER 700 SUCCESSFUL CLIENTS**
- **MORE THAN 1000 SCAMS EXPOSED**
- **MILLIONS SAVED IN POTENTIAL FRAUD**
- **PARTNERSHIPS WITH TOP LAUNCHPADS, INFLUENCERS AND CRYPTO PROJECTS**
- **CONSTANTLY BUILDING TOOLS TO HELP INVESTORS DO BETTER RESEARCH**

SPYWOLF

CRYPTO SECURITY

Audits | KYCs | dApps
Contract Development

To hire us, reach out to
contact@spywolf.co or
t.me/joe_SpyWolf

FIND US ONLINE

SPYWOLF.CO

@SPYWOLFNETWORK

@SPYWOLFNETWORK

Disclaimer

This report shows findings based on our limited project analysis, following good industry practice from the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, overall social media and website presence and team transparency details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report.
While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER:
By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice.
No one shall have any right to rely on the report or its contents, and SpyWolf and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (SpyWolf) owe no duty of care towards you or any other person, nor does SpyWolf make any warranty or representation to any person on the accuracy or completeness of the report.
The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and SpyWolf hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, SpyWolf hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SpyWolf, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts, website, social media and team.
No applications were reviewed for security. No product code has been reviewed.